

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-101788

(43)Date of publication of application : 04.04.2003

(51)Int.Cl.

H04N 1/41

H03M 7/30

H03M 7/40

H04N 7/30

(21)Application number : 2001-292589

(71)Applicant : RICOH CO LTD

(22)Date of filing : 25.09.2001

(72)Inventor : KADOWAKI YUKIO
OTSUKI TAKASHI
SATO YUTAKA

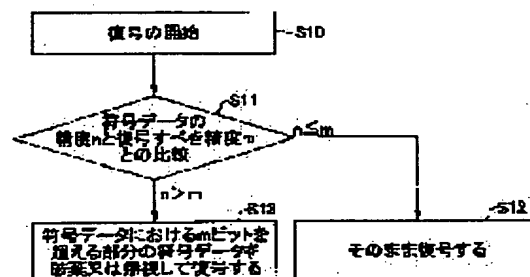
(54) DECODING METHOD, DECODING DEVICE, PICTURE PROCESSOR, PACKET HEADER GENERATING METHOD AND PACKET HEADER GENERATING DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To efficiently or quickly execute processing related with the encoding processing and/or decoding processing of picture data.

SOLUTION: In a method for decoding compression-encoded code data by dividing picture data in a frequency region expressed with n (n is a natural number which is not less than 2) bits into a plurality of partial picture data, the decoding processing is executed by abandoning the code data components of the part which exceeds m (m is a natural number which is smaller than n) bits of the code data (S13) so that m bit picture data can be extracted.

符号データを復号する復号装置の処理能力又は該復号装置が復号した画像データを表示する表示装置の表示能力に基づいて行う復号方法を決定するためのフローチャート



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開2003-101788

(P2003-101788A)

(43)公開日 平成15年4月4日(2003.4.4)

(51)Int.Cl. ⁷	識別記号	F I	テマコード(参考)
H 0 4 N 1/41		H 0 4 N 1/41	B 5 C 0 5 9
H 0 3 M 7/30		H 0 3 M 7/30	A 5 C 0 7 8
	7/40	7/40	5 J 0 6 4
H 0 4 N 7/30		H 0 4 N 7/133	Z

審査請求 未請求 請求項の数13 O L (全 17 頁)

(21)出願番号 特願2001-292589(P2001-292589)

(22)出願日 平成13年9月25日(2001.9.25)

(71)出願人 000006747

株式会社リコー

東京都大田区中馬込1丁目3番6号

(72)発明者 門脇 幸男

東京都大田区中馬込1丁目3番6号 株式
会社リコー内

(72)発明者 大槻 隆志

東京都大田区中馬込1丁目3番6号 株式
会社リコー内

(74)代理人 100070150

弁理士 伊東 忠彦

最終頁に続く

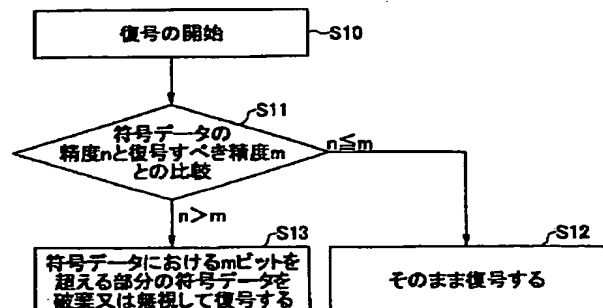
(54)【発明の名称】 復号方法、復号装置、画像処理装置、パケットヘッダ生成方法、パケットヘッダ生成装置

(57)【要約】

【課題】 画像データにおける符号化処理及び／又は復号処理に関する処理を効率的又は高速に行うことを目的とする。

【解決手段】 n (但し、 n は2以上の自然数)ビットで表わされる周波数領域の画像データを、複数の部分画像データに分割して、圧縮符号化された符号データの復号方法において、符号データにおける m (但し、 m は、 $m < n$ の自然数)ビットを超える部分の符号データ成分を破棄して復号処理を行い(S13)、 m ビットの画像データを取り出すことを特徴とする復号方法。

符号データを復号する復号装置の処理能力又は該復号装置が復号した画像データを表示する表示装置の表示能力に基づいて行う復号方法を説明するためのフローチャート



【特許請求の範囲】

【請求項1】 n （但し、 n は2以上の自然数）ビットで表わされる周波数領域の画像データを複数の部分画像データに分割し、該部分画像データのビットプレーン毎に圧縮符号化し、該圧縮符号化されたデータを、各部分画像データ毎に1つのパケットにして符号化した符号データを復号する復号方法であって、

前記 n ビットのうち、MSB側の m （但し、 m は、 $m < n$ の自然数）ビットのビットプレーンの符号データについて復号し、それ以外のビットのビットプレーンの符号データを破棄又は無視して復号処理を行い、 m ビットの画像復号データを取り出すことを特徴とする復号方法。

【請求項2】 前記 m は、符号データを復号する復号装置の処理能力又は該復号装置が復号した画像データを表示する表示装置の表示能力に基づいて定めることを特徴とする請求項1記載の復号方法。

【請求項3】 n （但し、 n は2以上の自然数）ビットで表わされる周波数領域の画像データを複数の部分画像データに分割し、該部分画像データのビットプレーン毎に圧縮符号化し、該圧縮符号化されたデータを、各部分画像データ毎に1つのパケットにして符号化した符号データを復号する復号装置であって、

前記 n ビットのうち、MSB側の m （但し、 m は、 $m < n$ の自然数）ビットのビットプレーンの符号データについて復号し、それ以外のビットのビットプレーンの符号データを破棄又は無視して復号処理を行う復号処理手段と、

m ビットの画像復号データを取り出す取り出し手段とを有することを特徴とする復号装置。

【請求項4】 前記 m は、符号データを復号する復号装置の処理能力又は該復号装置が復号した画像データを表示する表示装置の表示能力に基づいて設定することを特徴とする請求項3記載の復号装置。

【請求項5】 請求項3又は4記載の復号装置を有する画像処理装置。

【請求項6】 符号がJPEG2000が規定する符号であることを特徴とする請求項5記載の画像処理装置。

【請求項7】 パケットヘッダと圧縮データとの対により構成される複数のパケットと、該複数のパケット全体に係るヘッダであって、前記複数の全パケットの長さに関するデータを含む領域を有する上位ヘッダとを有する符号データを生成する符号データ生成方法において、
 先ず、各パケットの長さを求めるステップと、
 その後、該ステップにより求めたパケット長を用いて、前記パケットヘッダと前記上位ヘッダとを生成する上位ヘッダ生成ステップとを有することを特徴とする符号データ生成方法。

【請求項8】 JPEG2000の符号化データの作成にあたり、タイルパーツヘッダのタイルデータサイズを求めるために、先にパケットヘッダのデータ数を設定し

た後、パケットヘッダを再作成することを特徴とする符号データ生成方法。

【請求項9】 パケットヘッダと圧縮データとの対により構成される複数のパケットと、該複数のパケット全体に係るヘッダであって、前記複数の全パケットの長さに関するデータを含む領域を有する上位ヘッダとを有する符号データを生成する符号データ生成装置において、
 各パケットの長さを求めるパケット長検出手段と、
 前記パケット長検出手段により検出されたパケット長を積算する積算手段と、
 前記積算手段により積算されたパケット長を用いて、前記上位ヘッダを生成する上位ヘッダ生成手段とを有することを特徴とする符号データ生成装置。

【請求項10】 JPEG2000の符号化データの作成にあたり、タイルパーツヘッダのタイルデータサイズを求めるために、先にパケットヘッダのデータ数を設定した後、パケットヘッダを再作成することを特徴とする符号データ生成装置。

【請求項11】 請求項9又は10記載の符号データ生成装置を有する画像処理装置。

【請求項12】 JPEG2000による符号化を行う画像処理装置において、

パケットヘッダー・インフォーメーション

ゼロ・レンジス・パケット

コードブロック・インクルージョン

ゼロ・ビットプレーン・インフォーメーション

ナンバー・オブ・コーディング・パス

インクリーズ・オブ・コードブロック・インディケータ

レンジス・オブ・コードワード・セグメント

を各レイヤーにおける有効バイト数、ビットプレーン数、およびゼロビットプレーン数の情報から、ハードウェアにより、自動的にコーディングすることを特徴とする画像処理装置。

【請求項13】 請求項11又は12記載の画像処理装置において、

パイプライン処理を使ったパケットヘッダ生成装置を有することを特徴とする画像処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、復号方法、復号装置、画像処理装置、パケットヘッダ生成方法、パケットヘッダ生成装置に関する。

【0002】

【従来の技術】 図1にJPEG2000の符号化の流れを示す。JPEG2000の符号化においては、画像データ1は、2次元離散ウェーブレット変換部(DWT)10、量子化器11、係数モデリング部12、算術符号器13、符号形成部14の処理を経て、符号データ15が得られる。図2～4にこれらの処理を模式的に描いた図を示す。なお、係数モデリング部12及び算術符号器

13により、エントロピー符号化が行われる。

【0003】まず、入力される画像に対して複数レベルの2次元離散ウェーブレット変換を行う。この時、処理する画像をタイルと呼ばれる複数の矩形ブロックに分ける場合が多い。画像を複数のタイルに分割する場合は、後に続く2次元離散ウェーブレット変換、量子化、エントロピー符号化はこのタイルの単位で処理する。図2は画像データ1を 128×128 の大きさのタイルに分割し、レベル2の2次元離散ウェーブレット変換を行った場合の例を示す。 128×128 の画像データ(1タイル)は、レベル2の2次元離散ウェーブレット変換10により、大きさが 32×32 の4つのサブバンド2LL、2HL、2LH、2HHと大きさが 64×64 の3つのサブバンド1HL、1LH、1HHのウェーブレット係数データ21に変換される。JPEG2000では、ウェーブレット変換のフィルタ係数としては、可逆、非化逆の2通りの係数が規定されている。可逆・非化逆を統一的に扱う場合は、可逆のフィルタ係数が用いられ、より高いレート又は良好なひずみ特性を実現する場合は非化逆のフィルタ係数を用いる。

【0004】次に、ウェーブレット係数データに対して図3で示されるような式によりスカラー量子化が行われる。ここで、 $\text{sign}(a)$ はウェーブレット係数データaの符号、 $|a|$ はaの絶対値、 Δb はサブバンド毎に決められた量子化ステップ、 $\lfloor \cdot \rfloor$ はフロア関数を示す。ただし、可逆ウェーブレット係数を用いた場合は、このスカラー量子化の処理は実施されない。

【0005】次にウェーブレット変換後、あるいは量子化後の係数データを各サブバンド毎、あるいはサブバンドを複数の矩形領域に分割したコードブロック毎にエントロピー符号化していく。このエントロピー符号化は、後段の2値算術符号器13に与えるコンテキストを生成する係数モデリング部12での処理と、実際に符号化を行う算術符号器13での処理によりなされる。

【0006】図4は、各サブバンド毎にエントロピー符号化した場合におけるサブバンド2LLとその算術符号器13の処理を示す。サブバンド2LLは、 32×32 の画像サイズを有し、各点の画像は、 $(N+3)$ ビットで量子化されている。上面が量子化ビットのMSBで、下面が量子化ビットのLSBである。図4では、上から二つのプレーンには、量子化ビットが存在しない。このプレーンをゼロビットプレーンという。

【0007】次に、算術符号器13での処理手順を示す。

(1) 一つのサブバンド(コードブロック毎にエントロピー符号化する場合は、コードブロック)の係数データを符号+絶対値に変換し、係数絶対値は、ビットプレーンに分割してMSB側からビットプレーン毎にエントロピー符号化を行う。図4(A)では、2LLのサブバンドの係数絶対値をビットプレーンに分割した様子を示す。

(2) 上位から数えて初めて有効なビット(0で無いビット)が出現するビットプレーンまでは、ゼロビットプレーンであるのでエントロピー符号化は行わず、初めて0で無いビットが出現したビットプレーンから、エントロピー符号化を開始する。図4(A)の例では、上位2プレーンがゼロビットプレーンであり、3プレーン目に初めて0で無いビットが出現し、このビットプレーンNからビットプレーン0までのN+1枚のビットプレーンについてエントロピー符号化を行う。

(3) エントロピー符号化を行うビットプレーンに関しては、基本的に1つのビットプレーンを3回スキャンして、符号化する。この符号化のためスキャンする過程をコーディングパスとして、各パスは以下のように呼ばれている。

【0008】

- ① significance propagation pass
- ② magnitude refinement pass
- ③ cleanup pass

ビットプレーン上の各ビットは、特定の規則に従って分類され、そのビットの周辺ビットの状態から生成されたコンテキストを用いていずれかひとつのパスで符号化される。

【0009】図4(B)に示すように、最初に符号化するビットプレーンNは、cleanup passのみで符号化されるが、それ以降のビットプレーンは、それぞれ上に記した3つのパスで符号化される。すなわち、N+1個のビットプレーンが存在する場合、合計 $3N+1$ 回のコーディングパスにより符号化が行われる。

(4) 次に、算術符号器13で、上記の各パスで発生する符号化ビットとそれに対応するコンテキストにより、エントロピー符号を生成する。こうして生成されたエントロピー符号(MQ符号)は、図1に示す符号形成部14における処理で、最終のJPEG2000のビットストリームとしてまとめられる。

【0010】符号形成部14では、まず、先の算術符号器13で生成された各パス単位で発生した符号を、複数パス毎にまとめる。このまとめられた単位をレイヤと呼ぶ。図5では、 $3N+1$ 個のパス毎に算術符号器で生成された符号をレイヤ0からレイヤLまでのL+1個のレイヤにまとめている様子を示す。例えば、ビットプレーン2LLN、c、ビットプレーン2LLN-1、s、ビットプレーン2LLN-1、mをレイヤ0とし、・・・ビットプレーン2LL0、s、ビットプレーン2LL0、m、プレーン2LL0、cをレイヤLとしている。なお、レイヤの選定は、これに限らず実施することができる。

【0011】さらに、図6に示すように、サブバンド、レイヤ毎にまとめられた符号を並べて最終のビットストリームが生成される。図6では、2LLのレイヤ0、2LLのレイヤ1、・・・2LLのレイヤL、2HL、2

LH、2HHのレイヤ0、・・・2HL、2LH、2HHのレイヤ1、1HL、1LH、1HHのレイヤ0、・・・1HL、1LH、1HHのレイヤ1のビットストリームとなる。ビットストリームは、重要なデータ(優先順位の高いデータ)を先に送信するように構成される。なお、レイヤは、任意に設定できる。すべてを一つのレイヤとしてもよい。

【0012】ここで、サブバンド、レイア毎のまとまりをパケットと呼び、各パケットは、それぞれのパケットの情報を示すパケットヘッダーと先ほどの算術符号部で生成されたエントロピー符号より成る。例えば、2HL、2LH、2HHのレイヤ134は、2HL、2LH、2HHのレイヤ1のパケットヘッダー37と2HL、2LH、2HHのレイヤ1のエントロピー符号38から構成される。したがって、ビットストリームは、1又は複数のパケットにより構成される。

【0013】ここでは、入力画像データが単色である場合を示したが、複色色(コンポーネント)よりなる画像の場合、同様に各色(コンポーネント)毎にサブバンド毎、レイア毎のパケットを生成して規定された順番に並べることでJPEG2000のビットストリームを得ることができる。図7には、3色(コンポーネント0、1、2)より成る画像をJPEG2000で符号化した場合のビットストリームの例を示す。

【0014】JPEG2000の符号化方式は、いったん符号化されたビットストリームを復号することなく符号状態のまま再度圧縮して、必要な圧縮率を得られるという特徴がある。これはこれまでに説明したようにJPEG2000の符号は、パケットと呼ばれるコンポーネント(色成分)、サブバンド(解像度)、レイア毎の符号を組み合わせて構成されているため、一旦、符号化を行った後で、その符号の圧縮率が所望の圧縮率より悪ければ、復号した際の画質面からみて優先順位の低いパケットの符号データを順次破棄していくことで、圧縮率を上げる操作が可能であることによる。図7で示される符号は可逆(ロスレス)であるが、例えば図8に示すように、図7の符号の各コンポーネントにおいて、優先順位の低い1HL、1LH、1HHのレイア1のパケットを破棄することで、ロシー(非可逆)ではあるが、図7の符号をさらに圧縮した(圧縮率を上げた)符号を得ることができる。

【0015】

【発明が解決しようとする課題】JPEG2000で圧縮される画像データはさまざまな解像度を持っている。たとえば、一般的なテレビ画像であれば8ビット精度の画像データであるが、医療用のレントゲン写真などは16ビット精度のデータが必要になる。これは、医療用などでは、画像データをデータ保存用として扱い、後でさまざまな画像処理を施して疾患部分の抽出等を行うからである。

【0016】ところで、これまでは、画像表示デバイスがデジタルで8ビット程度の精度しか持っていない場合は、8ビットの精度で符号化して送信するか、又は、16ビットの表示装置でも、8ビットの表示装置でも表示できるように、符号化した符号データを用いていた。

【0017】例えば、前者では、表示装置の性能に合わせて、8ビットまでしか符号化しない符号化を行い、後者では、16ビットと8ビットの精度で符号化した符号データを用いていた。

【0018】しかしながら、表示装置の性能に合わせて符号化するのは、手間がかかり煩雑となる。また、この場合の符号データでは、16ビットの表示装置には不適切であり、汎用性がないという問題がある。

【0019】また、16ビットと8ビットの精度で符号化した符号データを用いる方式では、16ビットと8ビットの精度で符号化する処理と、少なくとも、16ビットと8ビットに対して、それぞれにヘッダが必要となり、処理に時間を要し、かつ、圧縮率が低下するという問題がある。

【0020】また、JPEG2000においては、タイル、レイア、コンポーネント、解像度レベルで表現された全ての圧縮画像データは、パケットヘッダとパケットデータ(圧縮データ)との対により構成されるパケットを含むコードストリームで符号化される。また、パケットデータの内容はパケットヘッダにより明記されている。ところで、このパケットヘッダの生成をマイクロプロセッサ等で処理するためには負荷が多く、他の処理に対する影響が大きいという問題がある。

【0021】本発明は、上記問題に鑑みなされたものであり、画像データにおける符号化処理及び/又は復号処理に関する処理を効率的又は高速に行うことを目的とするものである。

【0022】

【課題を解決するための手段】上記課題を解決するために、本件発明は、以下の特徴を有する課題を解決するための手段を採用している。

【0023】請求項1に記載された発明は、 n (但し、 n は2以上の自然数)ビットで表わされる周波数領域の画像データを複数の部分画像データに分割し、該部分画像データのビットプレーン毎に圧縮符号化し、該圧縮符号化されたデータを、各部分画像データ毎に1つのパケットにして符号化した符号データを復号する復号方法であって、前記 n ビットのうち、MSB側の m (但し、 m は、 $m < n$ の自然数)ビットのビットプレーンの符号データについて復号し、それ以外のビットのビットプレーンの符号データを破棄又は無視して復号処理を行い、 m ビットの画像復号データを取り出すことを特徴とする。

【0024】請求項1に記載された発明によれば、各部分画像データ毎に1つのパケットにして符号化して符号化しているので、データの圧縮率が高くなり、復号側に

おける復号の必要なビット数に関係なく符号化することができる。また、復号側では、 n ビットのうち、MSB側の m ビットのビットプレーンの符号データについて復号し、それ以外のビットのビットプレーンの符号データを破棄又は無視して復号処理を行うことにより、処理に必要なビット数だけを復号するので、すべてのビットについて復号する場合と比較して、処理負担が軽減される。

【0025】請求項2に記載された発明は、請求項1記載の復号方法において、前記 m は、符号データを復号する復号装置の処理能力又は該復号装置が復号した画像データを表示する表示装置の表示能力に基づいて定めることを特徴とする。

【0026】請求項2に記載された発明によれば、復号するビット数 m を、符号データを復号する復号装置の処理能力又は該復号装置が復号した画像データを表示する表示装置の表示能力に基づいて定めることにより、無駄な復号における処理を省くことができる。また、その結果、処理速度を向上することができる。

【0027】請求項3に記載された発明は、 n （但し、 n は2以上の自然数）ビットで表わされる周波数領域の画像データを複数の部分画像データに分割し、該部分画像データのビットプレーン毎に圧縮符号化し、該圧縮符号化されたデータを、各部分画像データ毎に1つのパケットにして符号化した符号データを復号する復号装置であって、前記 n ビットのうち、MSB側の m （但し、 m は、 $m < n$ の自然数）ビットのビットプレーンの符号データについて復号し、それ以外のビットのビットプレーンの符号データを破棄又は無視して復号処理を行う復号処理手段と、 m ビットの画像復号データを取り出す取り出し手段とを有することを特徴とする。

【0028】請求項4に記載された発明は、請求項3記載の復号装置において、前記 m は、符号データを復号する復号装置の処理能力又は該復号装置が復号した画像データを表示する表示装置の表示能力に基づいて設定することを特徴とする。

【0029】請求項3又は4記載の発明によれば、請求項1又は2記載の復号方法に適した復号装置を提供することができる。

【0030】請求項5に記載された発明は、請求項3又は4記載の復号装置を有する画像処理装置である。

【0031】請求項5記載の発明によれば、効率的に復号可能な請求項3又は4記載の復号装置を有する画像処理装置を提供することができる。

【0032】請求項6に記載された発明は、請求項5記載の画像処理装置において、符号化がJPEG2000が規定する符号化であることを特徴とする。

【0033】請求項6に記載された発明によれば、符号がJPEG2000が規定する符号であることにより、JPEG2000が規定する符号データを効率的に復号

することができる。

【0034】請求項7に記載された発明は、パケットヘッダと圧縮データとの対により構成される複数のパケットと、該複数のパケット全体に係るヘッダであって、前記複数の全パケットの長さに関するデータを含む領域を有する上位ヘッダとを有する符号データを生成する符号データ生成方法において、先ず、各パケットの長さを求めるステップと、その後、該ステップにより求めたパケット長を用いて、前記パケットヘッダと前記上位ヘッダとを生成する上位ヘッダ生成ステップとを有することを特徴とする。

【0035】請求項7に記載された発明によれば、パケットヘッダと圧縮データとの対により構成される複数のパケットと、該複数のパケット全体に係るヘッダであって、複数の全パケットの長さに関するデータを含む領域を有する上位ヘッダとを有する符号データを生成する符号データ生成方法において、先ず、各パケットの長さを求め、その後、該ステップにより求めたパケット長を用いて、パケットヘッダと上位ヘッダとを生成することにより、効率的に、符号データを生成することができる。

【0036】請求項8に記載された発明は、JPEG2000の符号化データの作成にあたり、タイルパーツヘッダのタイルデータサイズを求めるために、先にパケットヘッダのデータ数を設定した後、パケットヘッダを再作成することを特徴とする。

【0037】請求項8に記載された発明によれば、JPEG2000の符号化データの作成にあたり、タイルパーツヘッダのタイルデータサイズを求めるために、先にパケットヘッダのデータ数を設定した後、パケットヘッダを再作成することにより、JPEG2000の符号化データの生成を効率的に行うことができる。特に、JPEG2000の符号は、データ圧縮率は高いものの、非可逆符号で可変長符号であり、しかも、多くの画素を纏めて符号化するので、符号化処理及び復号処理に関する処理が複雑であり、効率的な符号化は大きな意義がある。

【0038】請求項9に記載された発明は、パケットヘッダと圧縮データとの対により構成される複数のパケットと、該複数のパケット全体に係るヘッダであって、前記複数の全パケットの長さに関するデータを含む領域を有する上位ヘッダとを有する符号データを生成する符号データ生成装置において、各パケットの長さを求めるパケット長検出手段と、前記パケット長検出手段により検出されたパケット長を積算する積算手段と、前記積算手段により積算されたパケット長を用いて、前記上位ヘッダを生成する上位ヘッダ生成手段とを有することを特徴とする。

【0039】請求項9に記載された発明によれば、各パケットの長さを求めるパケット長検出手段と、前記パケ

ット長検出手段により検出されたパケット長を積算する積算手段と、前記積算手段により積算されたパケット長を用いて、前記上位ヘッダを生成する上位ヘッダ生成手段とを有することにより、効率的に、符号データを生成することができる。

【0040】請求項10に記載された発明は、JPEG2000の符号化データの作成にあたり、タイルパートヘッダのタイルデータサイズを求めるために、先にパケットヘッダのデータ数を設定した後、パケットヘッダを再作成することを特徴とする符号データ生成装置である。

【0041】請求項10記載の発明によれば、請求項8記載の符号データ生成方法に適した符号データ生成装置を提供することができる。

【0042】請求項11に記載された発明は、効率的に符号データを生成することができる請求項9又は10記載の符号データ生成装置を有する画像処理装置である。

【0043】請求項11に記載された発明は、請求項9又は10に記載の符号データ生成装置を有する画像処理装置を提供することができる。

【0044】請求項12に記載された発明は、JPEG2000による符号化を行う画像処理装置において、パケットヘッダ・インフォメーション、ゼロ・レンジス・パケット、コードブロック・インクルージョン、ゼロ・ビットプレーン・インフォメーション、ナンバー・オブ・コーディング・パス、インクリーズ・オブ・コードブロック・インディケータ、レンジス・オブ・コードワード・セグメントを各レイヤーにおける有効バイト数、ビットプレーン数、およびゼロビットプレーン数の情報から、ハードウェアにより、自動的にコーディングすることを特徴とする。

【0045】請求項12に記載された発明によれば、パケットヘッダ・インフォメーション、ゼロ・レンジス・パケット、コードブロック・インクルージョン、ゼロ・ビットプレーン・インフォメーション、ナンバー・オブ・コーディング・パス、インクリーズ・オブ・コードブロック・インディケータ、レンジス・オブ・コードワード・セグメントを各レイヤーにおける有効バイト数、ビットプレーン数、およびゼロビットプレーン数の情報から、ハードウェアにより、自動的にコーディングすることにより、画像データに係る符号化処理を高速に行うことができる。

【0046】請求項13に記載された発明は、請求項11又は12に記載の画像処理装置において、パイプライン処理を使ったパケットヘッダ生成装置を有することを特徴とする。

【0047】請求項13に記載された発明によれば、パイプライン処理を使ってパケットヘッダ生成を行うことにより、多くのレジスタを削減したハードウェア構成とすることができる。

【0048】

【発明の実施の形態】次に、本発明の実施の形態について図面と共に説明する。なお、本実施の形態は、JPEG2000ベースにして、説明する。

【0049】図9に本発明が実施される1チップの符号化・復号器40の例を示す。図9の符号化・復号器40は、前・後処理部41、2次元離散ウェーブレット変換部(DWT)42、エントロピー符号・復号化部43、符号形成部44、PLL(Phase Locked Loop)回路45、CPU I/F部46、エントロピー符号化部43のワークメモリ47及びヘッダ処理部48から構成されている。前・後処理部41は、符号化時は画像データを画像データバスIMD19-0から得て、例えば、レベルシフトを行い、RGB(赤・緑・青)データをYCbCr(輝度成分・色差成分)データに変換する。また、復号時はその逆変換を行う。2次元離散ウェーブレット変換部(DWT/IDWT)42は、2次元離散ウェーブレット変換及びその逆変換を行う。エントロピー符号・復号化部43は、エントロピー符号化及びその逆変換を行う。符号形成部44は、符号化時は、符号形成を行い、生成した符号にヘッダ処理部48で生成したヘッダを付加して符号データバスCOD7-0によりビットストリームを有する符号データフォーマットを出力する。なお、ヘッダ処理部48は、メインヘッダ以外のタイルパートヘッダを自動的に生成する。なお、ヘッダ処理部48で、メインヘッダを生成するようにしてもよい。復号時は、符号データバスCOD7-0からビットストリームを有する符号データフォーマットを得て、符号化時と逆の処理を行う。PLL回路45は、符号化・復号器40のクロックを生成する。CPU I/F部46は、符号化・復号器40と外部に設けられたCPUとのインタフェースをとる。

【0050】この符号化・復号器40により、符号化時は、画像データバスIMD19-0から得た画像データを符号化して、符号データバスCOD7-0から、ビットストリームを有する符号データフォーマットを出力する。また、復号時は、符号データバスCOD7-0からビットストリームを有する符号データフォーマットを得て、復号して画像データバスIMD19-0へ画像データを出力する。

【0051】なお、図9において、IMD19-0は、画像データの入出力バスである。IMFRMは、入出力する画像データの有効期間を示す(アクティブ「L」)。IMMRDYは、クロック同期入出力モードのときは、外部が画像データを入出力できる状態にあることを示すレディー信号を示し、DMA転送モードのときは、外部からの画像データのアック信号としても機能する(アクティブ「L」)。IMJRDYは、クロック同期入出力モードのときは、符号化・復号器40が画像

データを入出力できる状態にあることを示すレディー信号を示し、DMA転送モードのときは、符号化・復号器40が出力する画像データのリクエスト信号として機能する(アクティブ「L」)。COD7-0は、クロック同期符号データの入出力バスである。COFRMは、入出力する符号データの有効期間を示す(アクティブ「L」)。COMMRDYは、外部が符号データを入出力できる状態にあることを示すレディー信号を示す(アクティブ「L」)。COJRDYは、符号化・復号器40が画像データを入出力できる状態にあることを示す(アクティブ「L」)。CPOUTは、PLLのマスタクロックとの位相差分の信号であり、VCOINは、LPF(Low Pass Filter)49を経た信号で、PLL45からの位相差分の信号でVCOを制御する信号である。MCLKは、PLLへのマスタクロックを入力する端子である。RESTは、符号化・復号器40のリセット端子を示す。INTは、設定に従って、割り込み信号を発生する(アクティブ「L」)。DACKは、画像データ、符号データをCPUバスを使用してDMA転送により入出力する際の符号化・復号器40へのデータアクノリッジ信号を示す(アクティブ「L」)。DREQは、画像データ、符号データをCPUバスを使用してDMA転送により入出力する際の符号化・復号器40からのデータリクエスト信号を示す(アクティブ「L」)。RDは、符号データの読み出し信号を示す(アクティブ「L」)。WEは、符号データの書き込み信号を示す(アクティブ「L」)。CSは、CPU/Fのチップセレクト信号を示す(アクティブ「L」)。AD9-0は、CPU/Fのアドバイスを示す。DATA31-0は、CPU/Fのデータバスを示す。

【0052】図10に、符号化・復号器40から符号データバスCOD7-0に出力される符号データフォーマットの例を示す。符号データは必ずメインヘッダーから始まる。メインヘッダーの始まりを示すコードはSOC(Start of Codestream)51である。この後にメインヘッダー内の情報が示されるmain(main header maker segment)52が、標準にしたがって記述される。また、図10に示すように、メインヘッダーに続いて、タイルの数分のタイルパーツヘッダーとビットストリームの組み合わせが続く。符号データフォーマットの最後を表すEOC(End of Codestream)で終わる。なお、ビットストリームは、図6で説明したように、パケット形式の信号である。

【0053】また、タイル毎にタイルパーツヘッダーが生成される。ビットストリーム560のタイルパーツヘッダーは、タイルパーツの開始を表すSOT53、タイルパーツヘッダーの内容を表すT0、TPO(Tile 0 header maker segment)54

0及びデータの開始を表すSOD(Start of Data)55から構成されている。同様に、ビットストリーム561のタイルパーツヘッダーは、SOT53、T1、TP1541、SOD55から構成されている。

【0054】図11にメインヘッダーの構成を示す。メインヘッダーは、SOC51に続いて、サイズを示すマーカーであるSIZ(Image and tile Size:必須)61から始まり、COD(Coding style default:必須)62、COC(Coding style Component:任意)63、QCD(Quantization default:必須)64、QCC(Quantization component:任意)65、RGN(Region of interest:任意)66、POC(Progression order change:任意)67、PPM(Packed Packet headers:任意)68、TLM(Tile-part lengths:任意)69、PLM(Packet lengths:任意)70、CRG(Component registration:任意)71及びCOM(Component:任意)72から構成されている。

【0055】SIZ61のマーカーは、取り扱う画像サイズ(イメージエリアの大きさ、タイルのサイズ(縦・横)、コンポーネントの数等)を規定する。その後は、順番は任意であるが符号化、復号に必要な情報が入っているCOD62のマーカーと量子化制御に必要な情報が入っているQCD64のマーカーが必須であり、残りのマーカーは任意である。このCOD62のマーカー内に元イメージの画像データビットの精度等が入っている。なお、PLM70は、必須ではないが、PLM70には、メインヘッダが先導するすべてのタイルパーツのパケット長等の情報が入れられる。

【0056】メインヘッダーの記述が完了すると、タイルパートヘッダーが始まる。タイルパートヘッダーの構成を図12に示す。タイルパートヘッダーは、SOT53から始まり、SOD55で終了する。SOT53とSOD55の間に、図12に示されているように、COD(任意)62、COC(任意)63、QCD(必須)64、QCC(任意)65、RGN(任意)66、POC(任意)67、PPT(Packed Packet headers、Tile-part header:任意)73、PLT(Packet lengths、Tile-part header:任意)74及びCOM(任意)72から構成されている。

【0057】なお、SOT53は、必須のヘッダであり、このヘッダには、タイルパートの長さ情報等が格納されている。したがって、タイルパートヘッダを生成するには、ビットストリームに含まれているパケットの長さ(=パケットヘッダの長さ+パケットデータの長さ)を知る必要がある。また、タイルデータ(=ビットスト

リーム56)はSOD55のマーカの次のバイトから出力される。すべての符号データの最後にはEOC57のマーカがつけられていて、このマーカですべてのコードの終了を示す。

【0058】次に、図11におけるメインヘッダのサイズマーカ61のフォーマットを説明する。図13にSIZマーカのフォーマットを示す。SIZマーカは0xFF51で示されるヘッダで始まる。LsizはSIZマーカのバイト数を示す。Rsizは符号器が適切にコードストリームをデコードできるかどうかを示す。XsizはX方向のリファレンスグリッド幅を示し、YsizはY方向のリファレンスグリッド高さを示す。XOsisは水平方向左側のイメージデータ先端からリファレンスグリッドの原点までのオフセット値を示し、YOsisは垂直方向上方のイメージデータ先端からリファレンスグリッドの原点までのオフセット値をしめす。XTsizはリファレンスグリッドに対応したタイルの幅を示し、YTsizはリファレンスグリッドに対応したタイルの高さを示す。XOTsizはリファレンスグリッドの原点から最初のタイルの開始位置までの水平方向オフセット値を示し、YOTsizはリファレンスグリッドの原点から最初のタイルの開始位置までの垂直方向のオフセット値を示す。Csizはイメージ内のコンポーネント数(一般的には色の数)をしめす。Ssiz_iはコンポーネントiのイメージデータビット精度を示す。このパラメータはコンポーネントの数だけ存在する。XRsiz_iはコンポーネントiの水平方向のリファレンスグリッドにおけるイメージデータの間隔を示す。YRsiz_iはコンポーネントiの垂直方向のリファレンスグリッドにおけるイメージデータの間隔を示す。

【0059】Ssiz_iを参照することにより、このコードストリームの精度を知ることができる。

【0060】次にCODマーカのフォーマットを説明する。CODマーカはCoding style defaultのことで符号化の対象となるイメージデータ、又はタイルデータのコーディングスタイル、階層のレベル、レイヤーの数などが示されている。CODマーカはメインヘッダの中には必須の項目であるが、タイルヘッダにも記述することができる。タイルヘッダにもCODマーカがある場合はタイルパートヘッダのCODマーカの情報が優先される。

【0061】CODマーカは図14に示されるように、5個のブロックで構成されている。CODブロックは、CODマーカの始まりを示すコードで0xFF52と決まっている。LcodブロックはCODマーカのバイト数を示す。

【0062】Scodブロックはすべてのコンポーネントのコーディングスタイルを示す。図15にコンポーネントのコーディングスタイルの各パラメータの値を示す。

す。

【0063】このように定義された符号データを復号するとき、最初にSIZマーカからイメージデータの基本情報を抽出する。このとき、SIZマーカには元データのビット精度がコンポーネント毎に示されている。元データが、たとえば、R(赤)、G(緑)、B(青)の16ビット精度の画像データであるときを考える。このとき、SIZマーカの値は、Ssiz₀=16, Ssiz₁=16, Ssiz₂=16になっている。

【0064】JPEG2000ではイメージデータを、図4で説明したように、ビットプレーン単位で処理することを特徴としている。その概念図を図16に示す。図16には、イメージデータをウェーブレット変換し、64×64のサイズのデータのビット長方向にデータを並べたあと、同じビット位置のデータ(64×64の値を16ビットで表現し、同じ重さのビット位置のデータ)だけを取り出し(これがビットプレーンである)、ビットプレーン単位で圧縮、伸張処理を行うことを特徴としている。圧縮伸張の方法は算術符号化(MQコード)を使用している。

【0065】なお、JPEG2000における符号化データの性質を利用して、復号を行う際、LSB側のコーディングパスのデータが有効であっても、それらをすべて無効(ZERO)にすることで復号データのビット長を最初から削減した状態で処理を行うことができる。

【0066】図17に示すように、JPEG2000ではデータをビットプレーン単位で圧縮、伸張する特徴を利用して、入力された符号データのうち、出力に必要なビット精度分だけMSB側からビットプレーンの復号を行い、残りのビットプレーンの符号データを破棄又は無視(図面上では、ビットを削除すると記載されている。)する操作を示す。この場合、各イメージデータのLSB成分がいくらか失われるが、元のビット精度が大きい場合は、一般的なディスプレイで表示する分に対しては問題無く表示できるだけの情報量を持っている。

【0067】具体的には、たとえば、X線写真で取られたデータを16ビットの精度でデジタルデータに変換し、JPEG2000のアルゴリズムを使用して可逆の圧縮符号化されたデータがあるとする。このデータは可逆復号によって完全に元の16ビットのデジタル画像データに復号することが可能で、ノイズ除去やフィルタリングなどを行うことで特定部分の疾患の検査に使用することができる。

【0068】しかし、この画像データを8ビット精度のディスプレイで表示する場合、16ビットのイメージ画像をすべて復元しても、表示は8ビットの精度しかできないことになる。この場合、本発明のように、たとえば8ビット分の復号機能を持つハードウェアを準備していれば、16ビットをすべて復号するよりも小さなハードウェアで16ビットデータのうち8ビット分の表示が可

能になる。

【0069】この場合、発明が解決しようとする課題で述べたように、表示装置の性能に合せて、16ビットと8ビットの精度で復号できるように、例えば、各ビット毎にパケット化して符号化することもできる。このようにすれば、16ビット以下の表示機能の表示装置のすべてに、対応した符号データをしかしながら、この場合は、16ものパケットヘッダが必要となり、オーバーヘッドが大きくなり、符号の圧縮率が低下する。

【0070】本実施の形態では、16ビットで一つのパケットにして、符号化し、復号するとき、9ビット以上の符号データ部分を破棄又は無視して、復号を行うことにより、8ビットの精度で復号し、8ビット精度の表示装置で表示する。

【0071】これを一般化して説明する。 n （但し、 n は2以上の自然数）ビットで表わされる周波数領域の画像データを複数の部分画像データに分割し、該部分画像データのビットプレーン毎に圧縮符号化し、該圧縮符号化されたデータを、各部分画像データ毎に1つのパケットにして符号化し、これを図18のフロー図に従って、復号する。

【0072】復号が開始された場合（S10）、符号データを復号する復号装置の処理能力又は該復号装置が復号した画像データを表示する表示装置の表示能力に基づいて定める符号データの精度 m と符号データの精度 n とを比較する。なお、符号データの精度 n は、図11におけるQCD64を参照することにより、得ることができる。

【0073】 $n \leq m$ の場合は、そのまま符号データを復号する（S12）。また、 $n > m$ の場合は、前記 n ビットのうち、MSB側の m ビットのビットプレーンの符号データについて復号し、それ以外のビットのビットプレーンの符号データを破棄又は無視して復号処理を行う（S11）。ステップ11の場合、実際には、復号に当たり、図11におけるQCD64に格納されているサブバンドにおける量子化のステップサイズを無視して、復号することになる。

【0074】ところで、符号データ生成をソフトウェアで構成すれば、柔軟性は高いものの、速度が低下する。特に、JPEG2000の符号のように、大量のデータを扱う場合に、速度の低下が顕著に表れる。

【0075】そこで、本実施の形態では、符号化処理及び／又は復号処理に関する処理を高速に行うために、装置をハードウェアで構成した。

【0076】図10で説明したように、符号データフォーマットは、メインヘッダ、タイルパーツヘッダとビットストリームよりなる複数の対及びEOC57から構成されている。また、タイルパーツヘッダには、必ず、タイルパーツの初めを示すSOT53が含まれている。このSOT53は、タイルパーツの初めを示すと共に、タ

イルパーツの長さ情報等が格納されている。したがって、タイルパーツヘッダを生成するには、ビットストリームに含まれているパケットの長さ（＝パケットヘッダの長さ＋パケットデータの長さ）を知る必要があり、その結果、符号データフォーマットを生成するには、ビットストリームに含まれているパケットの長さを知る必要がある。

【0077】次に、パケットヘッダ情報の生成例を説明する。パケットヘッダを生成する場合、必要な情報は

- ① ゼロビットプレーン数
- ② コーディングパス数
- ③ 各コードブロックのバイト長

等である。なお、ここでは、サブバンドを複数の矩形領域に分割したコードブロック毎にエントロピー符号化した場合について説明する。したがって、この場合では、パケットは複数のコードブロックから構成されている。

【0078】次に、図19を用いて、パケットヘッダの生成方法を説明する。

(1)パケットヘッダは、最初に、データの有無を確認する（S20）。これはパケット内の各コードブロックのバイト数で判断でき、パケット内のすべてのコードブロックのバイト数が0であればパケットにデータが無いことになる。パケットにデータがある場合は先頭に“1”がセットされる。パケットにデータがない場合は先頭に“0”がセットされるが、パケットはバイト（8ビット）単位で構成しなければならないので、パケットにデータがない場合は8ビットがすべて“0”のコードが生成される。

(2)次に、パケットの中のコードブロックの有無を確認する（S21）。コードブロックにデータがある場合は次のビットに“1”をセットする。コードブロックにデータがない場合は“0”がセットされる。このビットが“0”の場合、次のビットは次のコードブロックの有無を判断するビットとなる。コードブロックにデータがある場合、次にゼロビットプレーンの数をセットする。ゼロビットプレーンの数を表現する方法は、ゼロビットプレーンの数だけ“0”をセットした後“1”をセットする。

(3)次に、コーディングパスの数をセットする（S22）。コーディングパスのセットの方法は、図20に基づいて行われる。図20を説明すると、1から5までは単純な対応であり、2進化してコーディングパス数としている。6から36までは、9ビットのデータにおける上位4ビットは、“1111”固定であり、これで6以上を表し、下位5ビットは2進数を表し、この数字に6を加算したものがコーディングパス数になる。コーディングパス数が、37を超えると符合は16ビットで表現されるようになり、上位9ビットが“111111111”固定であり、これで37以上の数を表すようになり、下位7ビットの2進数データに37を加算した数

が、コーディングパスの数になる。エンコードの場合は、与えられたコーディングパス数からこれらの符号データを作成する。

(4) 次に、コードブロックのバイト数を定義する(S23)が、この場合、バイト数をできるだけ少ないビット数で表現するために、先に与えられたコーディングパス数を使用し、また、Lブロックというビットを付加している。バイト数を表現するビット長は、コーディングパス数の2の対数表現した整数分に3を加算した数で与えられる。しかし、この数字でもバイト数を表現するのに不足する場合はLブロックを言うビットを付加し、バイト数を表現するのに必要なビット数を補うようにしている。Lブロックの表現方法は必要な数だけ“1”を連続してセットし、その後で“0”をセットする。この後で、バイト数を2進数でセットする。

(5) 以上で、ひとつのコードブロックの packets へッダが生成される。ひとつの packets へッダの中に複数のコードブロックがある場合、先のコードブロックのバイト数の後に続いて次のコードブロックの有無を調べるビットがセットされる。最後のコードブロックのバイト数の場合は、packets へッダの大きさは8の倍数になっているので、8の倍数になるまで“0”をセットする(S24)。

【0079】次に、例をあげて説明する。ひとつの packets 内にコードブロックがひとつだけしかなく、ゼロビットプレーン数が3、コーディングパス数が13、バイト数が2進数101011101の場合、packets へッダは、

1 1 0001 111100111 1110 101011101 0000

になる。最初の“1”は packets にデータがあることを示し、次の“1”はコードブロックにデータがあることを示し、次の“0001”はゼロビットプレーン数が3であることを示し、次の“111100111”はコーディングパス数が13であることを示し、次の“1110”はLブロックが3であることを示し、次の“101011101”はコードブロックのバイト数を示し、最後の“0000”は packets の最後なので packets へッダが8の倍数になるように補間したデータである。

【0080】なお、コーディングパス数が“111100111”で13になるのは、上位4ビットが“1111”で+6を表し、下位5ビットが“00111”で2進数の7であるので $7+6=13$ になる。また、Lブロックが“1110”であるのはバイト数を表現するために9ビット必要で、コーディングパス数が13であるので $\log_2 13=3$ に、3を加算しても9ビットを表現するには3ビット不足しているために、Lブロックで3ビット補間させるためである。

【0081】次に、JPEG2000が規定する符号化のように、packets へッダと圧縮データとの対により構成される複数の packets と、該複数の packets 全体に関

する packets へッダであって、前記複数の全 packets の長さに関するデータを含む領域を有する上位 packets へッダとを有する符号データを生成する符号データ生成方法において、先ず、各 packets の長さを求めるステップと、その後に、該ステップにより求めた packets 長を用いて、前記 packets へッダと前記上位 packets へッダとを生成する上位 packets ステップとを有する符号データ生成手順について、図21を用いて説明する。

(1) 先ず、各 packets の長さを求める(S31)。

(2) 次に、前記ステップにより求めた packets 長を用いて、タイルパーツへッダにおけるタイルサイズデータをセットする(S32)。

(3) その後で、各 packets へッダ及びタイルパーツへッダを生成する(S33)。(4) ビットストリームと生成されたへッダを組み合わせて符号データフォーマットを生成する(S34)。

【0082】次に、packets ・ヘッダー・インフォメーションにおける各情報を生成して、packets へッダを生成する構成を図22を用いて説明する。

【0083】図22の構成は、各レイヤーにおける有効バイト数情報格納レジスタ51、各レイヤーにおけるビットプレーン数情報格納レジスタ52、ゼロ・ビットプレーン情報格納レジスタ53、ゼロ・レンジ判定器54、ビット・プレーン判定器55、有効レイヤー検出器56、コーディング・パス数算出器57、log演算器58、ビット・レンジ数判定器59、タグ・トリ符号化器60、ゼロ・レンジ・packets 生成器61、コードブロック・インクルージョン生成器62、ゼロ・ビットプレーン・インフォメーション生成器63、ナンバー・オブ・コーディング・パス生成器64、インクリーズ・オブ・コードブロック・インディケータ生成器65、レンジ・オブ・コードワード・セグメント生成器66、エクストラ・ゼロ・ビット挿入器67から構成されている。

【0084】なお、ここで、タグ・トリとはハイアラキにおける正の整数による2次元配列を表現する方法で、このトリの全てのノードにおいて、それより下のノード(最大4まで)における最小整数が記録されているものである。

【0085】次に、packets ・ヘッダー・インフォメーションにおける各情報を生成するための手順を、図22を用いて詳細に説明する。

(1) ゼロ・レンジ・packets 生成

① 各レイヤーにおける有効バイト数情報格納レジスタ51に保存されている有効バイト数が“0”か“0でない”ことを判定する。

【0086】② 次に各レイヤー全ての有効バイト数が“0”であった場合、ゼロ・レンジ・packets を“0”とし、本 packets をゼロ・レンジとする。

【0087】③ ひとつ以上のレイヤにおいて有効バイ

ト数が“0でない”場合、ゼロ・レングス・パケットを“1”とし、本パケットをノンゼロ・レングスとする。

(2) コードブロック・インクルージョン生成

① 各レイヤーにおけるビットプレーン数情報格納レジスタ53に保存されているビットプレーン数情報から各レイヤーにおいてビットプレーンが存在するか否かを判定する。つまり、コードブロックが含まれているか否かを判定する。

【0088】② 在する場合、コードブロック・インクルージョンは“1”となり、存在しない場合、コードブロック・インクルージョンは“0”となる。

【0089】③ 同じコードブロックが既に前のパケットに含まれていた場合はタグ・トリ符号化が行われ、それ以外は“1”となる。

(3) ゼロ・ビットプレーン・インフォメーション生成

① ゼロ・ビットプレーン情報格納レジスタ53に保存されているゼロ・ビットプレーン情報からゼロ・ビットプレーン数を確認する。

【0090】② ードブロックが最初に含まれていた場合、上で確認したゼロ・ビットプレーン数をタグ・トリ符号化を行いゼロ・ビットプレーン・インフォメーションとする。

(4) ナンバー・オブ・コーディング・パス生成

① 各レイヤーにおけるビットプレーン数情報格納レジスタ53に保存されている情報から、最初に有効ビットプレーンが存在するレイヤーの検出を行う。

【0091】② 上記①で検出されたレイヤーにおいて最初のコーディング・パスを、以下の式で算出する。

$$【0092】1 + (3 \times N_{bp})$$

(ここで N_{bp} は、最初に有効ビットプレーン有りと判定されたレイヤーにおける有効ビットプレーン数である。)

③ 以降のレイヤーにおけるコーディング・パス数の計算は以下の計算式で算出される。

$$【0093】3 \times N_{bp}$$

④ 上記②、③で算出されたコーディング・パス数をナンバー・オブ・コーディング・パスとして生成する。

(5) インクリーズ・オブ・コードブロック・インディケータ生成

① (4)で算出されたコーディング・パス数において以下の計算を行う。

$$【0094】\log_2 (\text{コーディング・パス数})$$

② 上記計算結果と L_{block} によりレングス・オブ・コードワードを表すのに必要なビット数を以下の計算により算出する。

【0095】ビット数 = L_{block} + ①の算出値

(ここで L_{block} は状態変数である)

ここで求めたビット数で表せるバイト数範囲でコードワ

ード数が納まる場合はこのインクリーズ・オブ・コードブロック・インディケータは、“0”となり、レングス・インディケータはアンチェンジドとなる。

【0096】コードワード数がこの範囲に納まらない場合は、納めるのに必要な追加ビットを付加する必要がある、追加ビット数をインクリーズ・オブ・コードブロック・インディケータとして符号化する。

(6) レングス・オブ・コードワード・セグメント生成
① 各レイヤーにおける有効バイト数情報格納レジスタ51に保存されている有効バイト数を(5)で決定されたビット数で表現することによりレングス・オブ・コードワードとする。

(7) 以上(1)から(6)までで生成されたビット情報をバイト単位でパックするにあたり、そのバイト値が $0xFF$ であった場合、次のバイトのMSBにはエクストラ・ゼロ・ビットとして“0”ビットを自動的に挿入する。

(8) 以上(1)から(7)までで生成された情報をJPEG2000パケットヘッダーフォーマットに従い配置することによりパケット・ヘッダー・インフォメーションとして自動的に生成する。

【0097】なお、図22に示されているように、パケットヘッダを生成する構成を、ハードウェアで構成することにより、画像データに係る符号化処理を高速に行うことができる。

【0098】また、画像処理装置をパイプライン処理を使って、例えば、パケットヘッダ生成を行うことにより、多くのレジスタを削減したハードウェア構成とすることができる。

【0099】なお、上記実施の形態では、SOTに必要な、タイルパーツヘッダのタイルデータサイズを求めるために、先にパケットヘッダのデータ数を設定した後、パケットヘッダを再作成するように説明したが、このように求めたタイルデータサイズをメインヘッダのTLMで用いてもよい。

【0100】また、上記実施の形態で求めたパケットの長さ情報を、メインヘッダのPLM及びタイルパーツヘッダのPLTで用いてもよい。

【0101】本発明は、JPEG2000で高ビット精度で圧縮された画像データを、ディスプレイで表示するのに必要な低ビット精度で復号するのに適した画像復号装置を提供することができる。

【0102】本発明は、このパケットヘッダ生成処理をハードウェアで行わせることにより、高速に処理を行い、かつマイクロプロセッサ等の負荷を低減させることができる。

【0103】

【発明の効果】上述の如く本発明によれば、画像データにおける符号化処理及び／又は復号処理に関する処理を効率的又は高速に行うことができる。

【図面の簡単な説明】

【図1】JPEG2000の符号化の流れを説明するための図である。

【図2】画像データ1を128×128の大きさのタイルに分割し、レベル2の2次元離散ウェーブレット変換を行った場合の例を説明するための図である。

【図3】スカラー量子化の式である。

【図4】各サブバンド毎にエントロピー符号化した場合におけるサブバンド2LLとその算術符号化器の処理を説明するための図である。

【図5】3N+1個のパス毎に算術符号器で生成された符号をレイヤ0からレイヤLまでのL+1個のレイヤにまとめている様子を示す図である。

【図6】ビットストリームを説明するための図である。

【図7】3色（コンポーネント0、1、2）より成る画像をJPEG2000で符号化した場合のビットストリームの図である。

【図8】図7の符号の各コンポーネントの1HL、1LH、1HHのレイアルの packets を破棄したビットストリームの図である。

【図9】符号化・復号器の構成例を説明するための図である。

【図10】符号化・復号器から出力される符号データフォーマットの例を説明するための図である。

【図11】メインヘッダーの構成例を説明するための図である。

【図12】タイルパートヘッダーの構成例を説明するための図である。

【図13】SIZマーカーのフォーマットを説明するための図である。

【図14】CODマーカーのフォーマットを説明するための図である。

【図15】コーディングスタイルの各パラメータを説明するための図である。

【図16】JPEG2000におけるビットプレーン単位で処理する概念を説明するための図（その1）である。

【図17】JPEG2000におけるビットプレーン単位で処理する概念を説明するための図（その2）である。

【図18】符号データを復号する復号装置の処理能力又は該復号装置が復号した画像データを表示する表示装置の表示能力に基づいて行う復号方法を説明するためのフロー図である。

【図19】パケットヘッダーの生成方法を説明するためのフロー図である。

【図20】コーディングパスの符号データを説明するための図である。

【図21】符号データフォーマットの生成手順を説明するための図である。

【図22】パケットヘッダーを生成する構成例を説明するための図である。

【符号の説明】

- 1 画像データ
- 10 DWT部
- 11 量子化器
- 12 係数モデリング部
- 13 算術符号器
- 14 符号形成部
- 15 符号データ
- 20 1タイルの画像データ
- 37 パケットヘッダー
- 38 エントロピー符号
- 40 符号化・復号器
- 41 前・後処理部
- 42 DWT/IDWT
- 43 エントロピー符号・復号部
- 44 符号形成部
- 45 PLL
- 46 CPU I/F
- 47 ワークメモリ
- 48 ヘッダー処理部
- 49 LPF

【図1】

JPEG2000の符号化の流れを説明するための図



【図14】

CODマーカーのフォーマットを説明するための図



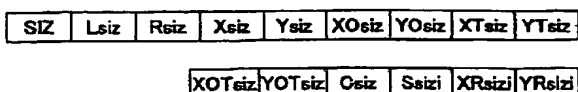
【図3】

スカラー量子化の式

$$q = \text{sign}(a) \times \lfloor |a| / \Delta b \rfloor$$

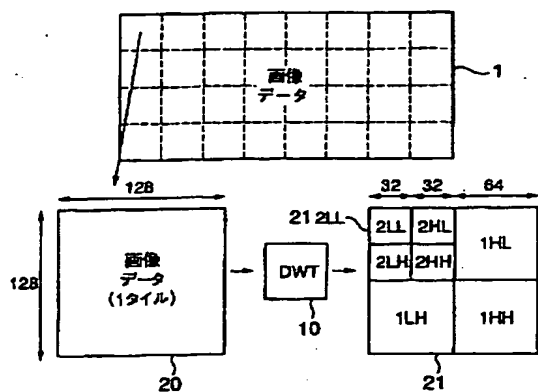
【図13】

SIZマーカーのフォーマットを説明するための図



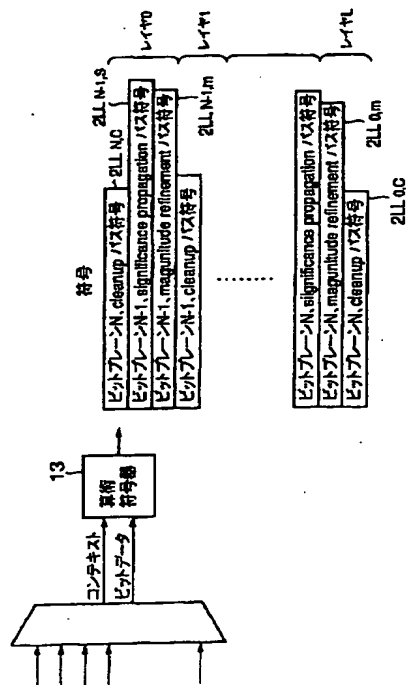
【圖 2】

画像データ1を128×128の大きさのタイルに分割し、レベル2の2次元離散ウェーブレット変換を行った場合の例を説明するための図



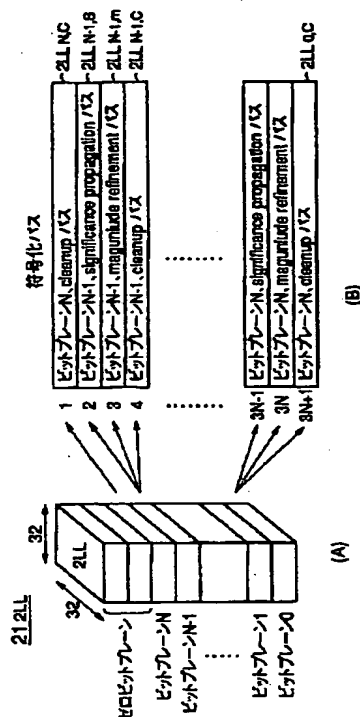
【图5】

3N+1個のバス毎に算術符号器で生成された符号をレイヤ0からレイヤLまでのL+1個のレイヤにまとめている様子を示す図



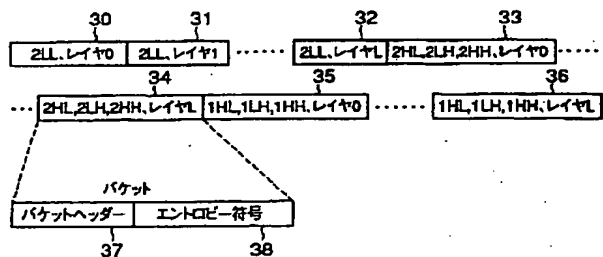
【图4】

各サブバンド毎にエントロピー符号化した場合におけるサブバンド211とその算術符号化器の処理を説明するための図



【図6】

ビットストリームを説明するための図



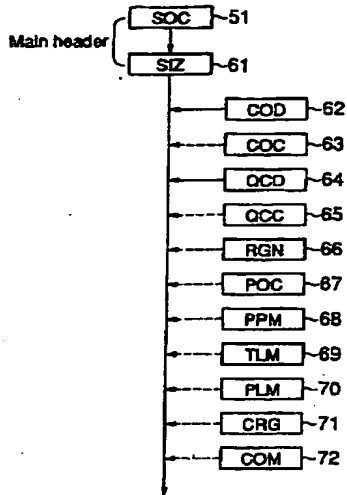
【图 15】

コーディングスタイルの各パラメータを説明するための図

Parameter	Size(Bits)	Value
OOD	16	0xFF52
Lcod	16	12~45
Scod	8	省略
SGood	32	省略
SPcod	variable	省略

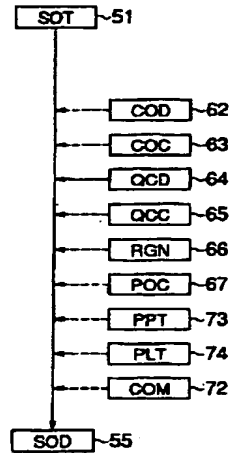
【図11】

メインヘッダーの構成例を説明するための図



【図12】

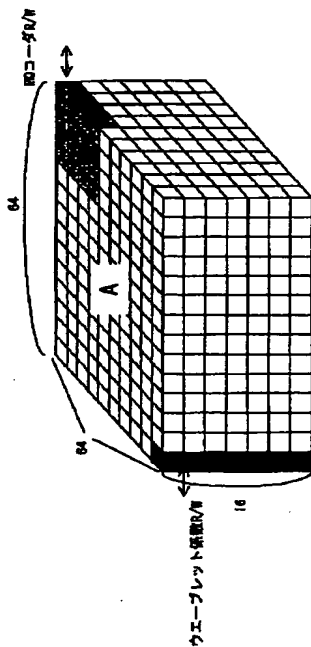
タイルパートヘッダーの構成例を説明するための図



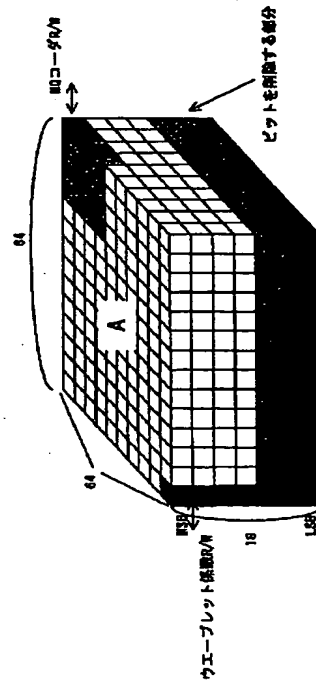
【図17】

【図16】

JPEG2000におけるビットプレーン単位で処理する概念を説明するための図(その1)

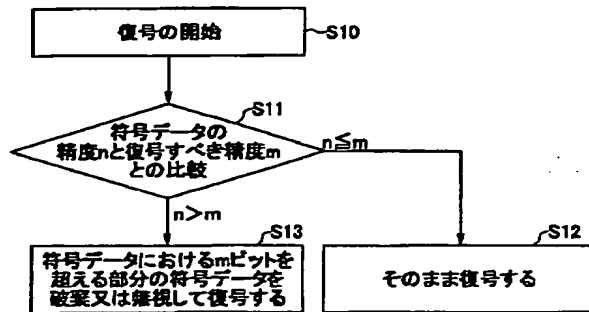


JPEG2000におけるビットプレーン単位で処理する概念を説明するための図(その2)



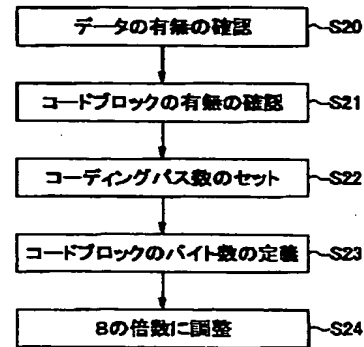
【図18】

符号データを復号する復号装置の処理能力又は該復号装置が復号した画像データを表示する表示装置の表示能力に基づいて行う復号方法を説明するためのフローチャート



【図19】

バケットヘッダの生成方法を説明するためのフロー図



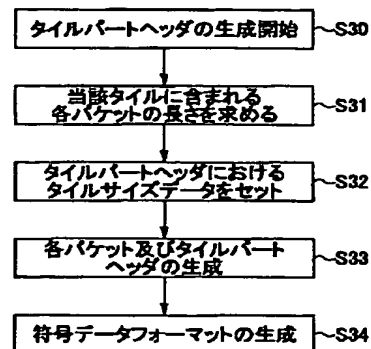
【図20】

コーディングパスの符号データを説明するための図

コーディングパスの数	コーディングパスの符号データ
1	0
2	10
3	1100
4	1101
5	1110
6~36	111100000~111111110
37~164	1111111110000000 ~1111111111111111

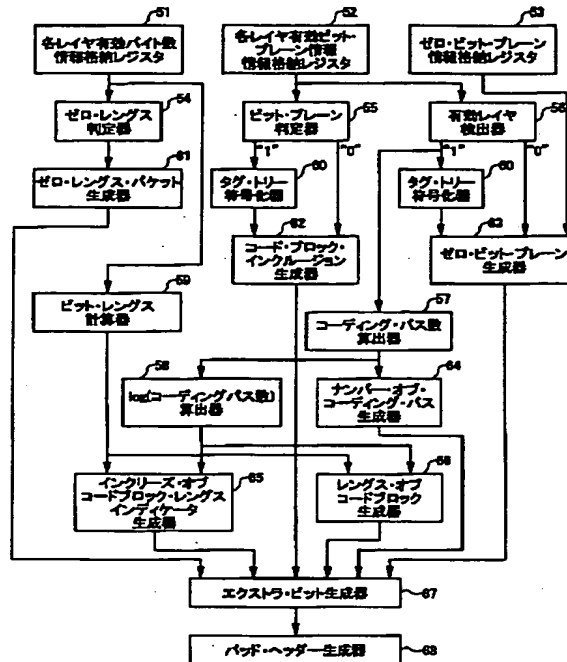
【図21】

符号データフォーマットの生成手順を説明するための図



【図22】

パケットヘッダを生成する構成例を説明するための図



フロントページの続き

(72)発明者 佐藤 豊
東京都大田区中馬込1丁目3番6号 株式
会社リコー内

Fターム(参考) 5C059 KK11 MA00 MA24 MA35 MC11
MC38 ME11 PP01 PP15 RB02
RC14 RC24 RC28 SS23 TA43
TA49 TB17 TC47 UA05 UA09
UA15 UA29 UA30
5C078 AA04 BA21 BA53 CA25 CA31
DA00 DA01 DA02 DB19 EA00
5J064 AA02 AA03 BA09 BA10 BA16
BB05 BC02 BC16 BD01